



Universidade do Porto
Faculdade de Engenharia
FEUP

Eficiência Energética em Redes de Sensores sem Fios

Relatório Intercalar

Agentes e Inteligência Artificial Distribuída
2015/2016

4.º ano do Mestrado Integrado em Engenharia Informática e Computação

Grupo T11_3:

Filipe Ribeiro – 201104129 – ei11141@fe.up.pt

Henrique Ferrolho – 201202772 – ei12079@fe.up.pt

João Pereira – 201104203 – ei12023@fe.up.pt

8 de Novembro de 2015

Índice

Índice

- 1 Enunciado
 - 1.1 Descrição do cenário
 - 1.2 Objetivos do trabalho
 - 1.3 Resultados esperados e forma de avaliação
2. Plataforma e Ferramentas
 - 2.1 Para que serve e principais características
 - JADE
 - REPAST
 - SAJaS
 - 2.2 Realce das funcionalidades relevantes para o trabalho
3. Especificação
 - 3.1 Identificação e caracterização dos agentes
 - 3.2 Protocolos de interação
 - COSA: Comportamento básico de um agente
 - 3.3 Faseamento do projecto
4. Recursos
 - 4.1 Bibliografia
 - 4.2 Software

1 Enunciado

1.1 Descrição do cenário

No âmbito da unidade curricular de Agentes e Inteligência Artificial Distribuída o grupo propôs-se a desenvolver um programa que possibilita a redução do gasto de energia numa Wireless Sensor Network (WSN).

O problema surge aquando da instalação de uma WSN em áreas inacessíveis ou remotas, em que é necessário o uso de algoritmos de percepção e comunicação que minimizem o gasto de energia. Este gasto é um contratempo devido ao elevado custo que a substituição e recarga das baterias dos sensores têm. Por outro lado, a informação sensorial transmitida tem de ser boa o suficiente de forma a permitir a tomada de decisões atempada. Há, portanto, que balancear a qualidade da informação com o consumo energético.

1.2 Objetivos do trabalho

O principal objetivo deste trabalho é a implementação de um algoritmo de formação de coligações, COSA, idealmente estabelecidas através de negociação peer-to-peer, e correr simulações de forma a testá-lo. A estrutura de coligação resultante irá depender do estado dos agentes e do ambiente a cada instante.

Como já foi referido, é necessário resolver o problema dos gastos de energia, portanto o algoritmo terá de ser o mais eficiente possível, mantendo a qualidade, de forma a reduzir o consumo energético. Para tal, utilizaremos técnicas de Inteligência Artificial, mais especificamente Sistemas Multi-Agentes.

1.3 Resultados esperados e forma de avaliação

A análise de resultados será uma componente bastante importante neste projeto, de forma a verificar a efetiva redução do gasto de energia e a eficiência da informação transmitida entre os sensores.

O **consumo de energia do sistema** será avaliado através de uma mediana dos valores de energia dos nós em cada instante, garantindo a informação do número de nós que ainda estão “vivos” na rede, $E(a_i) > 0$, e também o padrão de descarga das suas baterias. No que diz respeito à **qualidade da informação**, serão necessárias duas variáveis: o **erro** e a **entropia**.

O **erro** é medido com a diferença do valor atual observado do *sink node* para cada nó vivo e o atual valor de poluição. Isto possibilita saber o quão errado estará o *sink node* em relação à situação atual do rio. A fórmula do erro, no instante t , encontra-se a seguir:

$$e^t = \sum_{i \in N^t} \|xs_i^t - xp_i^t\|$$

onde N^t representa o conjunto de nós “vivos” no instante t ; xs_i^t , o valor conhecido pelo *sink node* para o nível de poluição no nó i no instante t e xp_i^t o nível atual de poluição do nó i , no instante t .

A **qualidade da informação** resulta da adição da entropia de cada nó da rede, “vivo” ou “morto”. De referir que a entropia de cada nó aumenta à medida que o tempo passa. A fórmula:

$$Information\ Quality = \sum_{i \in N^t} [H_i(t) = \ln(\sigma_i(t) \cdot \sqrt{2\pi e})]$$

onde H_i é o valor da **entropia** de cada nó, e σ_i é uma distribuição plana com uma função de decaimento:

$$\sigma_i(t) = [\sigma_{bot} \text{ if } t = t_i, \sigma_{bot} + \frac{e^{t-t_i}}{e^{t_{max}}} \cdot (\sigma_{top} - \sigma_{bot}) \text{ if } t \neq t_i]$$

onde σ_{bot} é uma variação do barulho gaussiano que o simulador adiciona a cada leitura de qualquer sensor, e σ_{top} é um valor alto de σ , podendo ser, por exemplo, $100\sigma_{bot}$, e t_{max} é atribuído com o triplo do valor do tempo de leitura de amostras (*sampling period*).

A forma de avaliação de resultados não terá nenhuma estratégia em vigor, isto é, implementar-se-á, para já, a versão com o algoritmo COSA e será ilustrada uma barra percentual com os valores finais esperados comparativamente a uma rede com amostragem aleatória. Se o tempo permitir o grupo tem estipulado duas estratégias a implementar junto com o algoritmo COSA: **Frequência de amostragem e Coerência**.

A primeira, **frequência de amostragem**, consiste em alterar a frequência com que são tiradas as amostras. Deste modo conseguimos aumentar a capacidade de reação da rede e os agentes detetam mudanças ao ambiente mais cedo.

A segunda, **coerência**, verifica se a liderança de um agente está coerente com os últimos valores amostrados com os atuais membros da coligação.

2. Plataforma e Ferramentas

2.1 Para que serve e principais características

JADE

Esta estrutura *open source* tem como objetivo simplificar o desenvolvimento de sistemas multi-agentes, de acordo com as especificações da *FIPA (Foundation for Intelligent Physical Agents)*. O *JADE* tem um sistema multi-agente distribuído, isto é, os agentes apenas podem agir dentro do seu domínio. Cada agente em *JADE* tem *behaviours*, que são tarefas a executar. Estas podem ser executadas concorrentemente e podem ser agendadas de uma forma cooperativa.

O *JADE* permite a mobilidade e *cloning* de agentes.

REPAST

O *REPAST* é utilizado para fazer simulações utilizando agentes. Ao contrário do *JADE*, não serve para desenvolver Sistemas Multi-agentes nem é compatível com as especificações da *FIPA*. Com o *REPAST* é possível fazer colheita de dados e consequente exposição e é possível ver a interação entre os agentes durante as simulações feitas.

SAJaS

O *SAJaS* será utilizado pela necessidade de realizar testes durante o desenvolvimento do Sistema Multi-agentes. Servirá como ligação entre a simulação e o desenvolvimento do sistema.

2.2 Realce das funcionalidades relevantes para o trabalho

As vantagens da *framework JADE* neste trabalho é fornecer abstrações como *Agent* e *Behaviour*, comunicação *peer-to-peer* entre agentes e mecanismo de descoberta, como o *publish-subscribe*, que possibilita aos agentes encontrarem-se mutuamente. *JADE* possui algumas funcionalidades adicionais como mobilidade de agentes, tolerância a falhas, integração com *web-services* e diversas ferramentas gráficas que facilitam a interação do utilizador com a aplicação.

Por outro lado, *Repast+SAJaS*, fornece a criação, análise e experimentos com cenários populados por agentes que interagem de formas não triviais, assim como visualização de resultados em gráficos e as interações entre os agentes.

3. Especificação

3.1 Identificação e caracterização dos agentes

O algoritmo COSA implica que os nós da rede - os **agentes** do sistema multi-agentes deste projeto - tenham um comportamento padrão de amostragem diferente do normal. O algoritmo implica que os agentes sejam **autónomos, proativos e reativos**. De acordo com o material utilizado nas aulas teóricas de AIAD, este tipo de arquitetura enquadra-se na de **agentes reativos simples**.

O comportamento dos agentes descrito no parágrafo acima é possível através de um protocolo de negociação simples, e de duas funções de modelação de relações entre os nós: **aderência e liderança**.

Os resultados destas funções de modelação de relações determinam o diálogo assíncrono que se dá enquanto os agentes estão a negociar entre eles. Por sua vez, os resultados dessas negociações modificam as funções de modelação de relações.

3.2 Protocolos de interação

Quando um nó faz uma amostragem do ambiente à sua volta, envia o valor observado para os seus vizinhos. Quando um nó recebe uma amostra de um dos seus vizinhos, usa essa informação para avaliar a pertinência de formar um grupo. Se essa avaliação for positiva, o agente comunica o resultado ao nó que lhe enviou a amostra, e assume o papel de líder. Se ambos concordarem, o agente assume definitivamente o papel de líder, enquanto que o outro assume o papel de dependente. O agente dependente pode hibernar e interromper tanto o seu processo de amostragem, bem como a comunicação com o *sink node*.

Decorrido algum tempo, as negociações simples entre agentes vizinhos acabam por originar uma coligação (através de uma abordagem *bottom-up*) e seleccionar o papel mais adequado nessa coligação.

A organização dos nós em grupos bem estruturados traduz-se numa poupança de energia devido à poupança de transmissões de longa distância: aquelas que os nós dependentes não têm que fazer.

COSA: Comportamento simplificado de um agente

```
while (bateria > 0) {  
  
    recolheAmostra();  
  
    atualizaModelo();  
  
    atualizaRelaçãoComVizinhos();  
  
    atualizaRedeDeNós();  
  
}
```

3.3 Faseamento do projecto

O grupo planeia dividir o desenvolvimento do projecto em **6 fases**.

Fase 1

Implementar o algoritmo COSA.

Fase 2

Testar e analisar o comportamento de um só agente com o código desenvolvido. É esperado que este agente assuma o papel de líder, uma vez que é o único existente.

Fase 3

Testar e analisar o comportamento de dois agentes. É esperado que um deles assuma o papel de líder, e que o outro assuma o papel de dependente.

Fase 4

Repetir o processo descrito nas fases 2 e 3 com múltiplos agentes.

Fase 5

Implementar e analisar os resultados de uma das estratégias descritas no artigo de apoio a este projecto.

Fase 6

Caso ainda seja possível, nesta fase o grupo irá implementar e comparar os resultados da estratégia restante do artigo de apoio.

4. Recursos

4.1 Bibliografia

- María del Carmen Delgado; Carles Sierra "A Multi-agent Approach to Energy-Aware Wireless Sensor Networks Organization", Agreement Technologies - Second International Conference, AT. Lecture Notes in Computer Science, vol. 8068, Beijing, China, Springer, pp. 32-47, 01/08/2013.

4.2 Software

O projecto em questão será desenvolvido no sistema operativo Ubuntu, através do IDE [Eclipse](#), com as *frameworks* [JADE](#), [Repast](#) e [SAJaS](#).